

2
X-692-72-92

NASA 65859

INTERACTIVE GRAPHICS SYSTEM FOR IBM 1800 COMPUTER

T. P. CARLETON
D. R. HOWELL
W. H. MISH

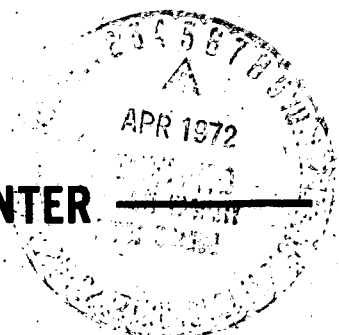
MAY 16 1973

MARCH 1972

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
Springfield, Va. 22151



GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND



(NASA-TM-X-65859) INTERACTIVE GRAPHICS
SYSTEM FOR IBM 1800 COMPUTER T.P.
Carleton, et al (NASA) Mar. 1972 55 p
CSCL 09B

N72-20182

Unclas
22346

G3/08

(CATEGORY)

55

INTERACTIVE GRAPHICS SYSTEM
FOR IBM 1800 COMPUTER

by

T. P. Carleton

D. R. Howell*

W. H. Mish

Laboratory for Extraterrestrial Physics
NASA Goddard Space Flight Center
Greenbelt, Maryland

March 1972

*LtJG USNR, on detached duty to NASA/GSFC

TABLE OF CONTENTS

	PAGE
ABSTRACT	-
1.0 Hardware and Software Descriptions	2
1.1 Hardware Description	2
1.2 Software Description	8
1.2.1 Software Concepts	8
1.2.1.1 Pictures	8
1.2.1.2 Dynamic Pictures	9
1.2.1.3 Function Matrix and Photopen	10
1.2.2 Software Definitions	12
1.2.2.1 Display	12
1.2.2.2 Picture	12
1.2.2.3 Origin	12
1.2.2.4 Raster Unit	12
1.2.2.5 Picture ID	12
1.2.2.6 Intensity	13
1.2.2.7 Density	13
1.2.2.8 Picture Pointer Table (PPT)	13
1.2.2.9 Data Table	13
1.2.2.10 Free Table List (FTL)	13
1.2.2.11 Free Disk Area (FDA)	13
1.2.2.12 Data Table Area	14
1.2.2.13 INSKEL COMMON	14
1.2.3 Subroutines Directly Available to the Application Programmer	16
1.2.3.1 ALPHZ	16
1.2.3.2 CPICZ	17
1.2.3.3 DEN2Z	17
1.2.3.4 DLARY	17
1.2.3.5 DLETZ	18
1.2.3.6 EXCRT	19
1.2.3.7 FACTZ	19
1.2.3.8 FLPTZ	19
1.2.3.9 FPICZ	20
1.2.3.10 GETZZ	20
1.2.3.11 GTXYZ	20
1.2.3.12 INDXZ	20
1.2.3.13 INITZ	20
1.2.3.14 INTGZ	21
1.2.3.15 INT1Z	21
1.2.3.16 INT2Z	21
1.2.3.17 INT3Z	21
1.2.3.18 LIPNP	22

	PAGE
1.2.3.19 MOVEZ	22
1.2.3.20 OFFZ	22
1.2.3.21 ONZZ	22
1.2.3.22 PLOTZ	23
1.2.3.23 PUTZZ	23
1.2.3.24 RSETZ	23
1.2.3.25 RSCLZ	23
1.2.3.26 SCALZ	24
1.2.4 System Subroutines	25
1.2.4.1 COMDP	25
1.2.4.2 DPICZ	25
1.2.4.3 ERRZ	25
1.2.4.4 FDSAZ	25
1.2.4.5 FMTZ	25
1.2.4.6 GTABZ	25
1.2.4.7 LITPN	26
1.2.5 Examples	27
1.2.5.1 Example of Data Editing Via Interactive Software	27
1.2.5.2 An Example of the User Error Recovery Option	35
2.0 Software Internals	38
2.1 Description of Tables	38
3.0 References	44
4.0 Appendices	45
4.1 Appendix A - CRT Error Messages	45
4.2 Appendix B - Subroutine References	47
4.3 Appendix C - Subroutine Storage Requirements	48
4.4 Appendix D - Indexed Data Arrays	49
4.5 Appendix E - Steps for Writing a CRT Display Program	50

ABSTRACT

This paper describes a FORTRAN compatible software system that has been developed to provide an interactive graphics capability for the IBM 1800 computer.

The interactive graphics hardware consists of a Hewlett-Packard 1300A Cathode Ray Tube, Sanders Photopen, digital to analog converters, pulse counter and necessary interface. The hardware is available from IBM as several related RPQ's.

The software developed permits the application programmer to use IBM 1800 FORTRAN to develop a display on the cathode ray tube which consists of one or more independent units called pictures. The software permits a great deal of flexibility in the manipulation of these pictures and allows the programmer to use the photopen to interact with the displayed data and make decisions based on information returned by the photopen.

1.0 Hardware and Software Descriptions

1.1 Hardware Description

The Hewlett Packard (HP) 1300A Cathode Ray Tube (CRT) consists of a glass bottle like container housing a electron gun which emits electrons and an electron optical system to accelerate, deflect and focus the emitted electrons beam onto a phosphor screen. A part of the electron optical system are the horizontal (X) and vertical (Y) electrostatic deflection amplifiers which control the horizontal and vertical position of the focused beam of electrons on the phosphor screen. The electrons striking the phosphor cause the emission of light of some spectral distribution determined by the composition of the phosphor. In addition there is a Z axis amplifier that controls the current of the beam and thus controls the luminance (intensity) of the dots of light that make up the display. The HP 1300A uses aluminized, P31 phosphor with an 8x10 inch screen. P31 is a short persistence phosphor (it takes 38×10^{-6} sec. for the luminance (intensity) to fall to 10% of its initial value) and to avoid the sensation of flicker the display needs to be refreshed at a minimum of about 38 times per second. Flicker is the sensation of brightness variation caused when the refresh rate is too low.

Digital information to be displayed on this cathode ray tube are sent via a 1800 data channel to a modified 1856 analog output terminal. The modified 1856 consists, in part, of four 10 bit and one 3 bit digital to analog converters (DACs) which convert the digital information to the voltages for input to the X, Y and Z amplifiers of the cathode ray tube. The 5 DACs provide X, incremental X (ΔX), Y, incremental Y (ΔY) and Z (intensity) information to the cathode ray tube. Approximately 105,000

words/second can be transferred to the CRT when the intensity information is non zero. Approximately 285,000 words/second can be transferred when intensity information is blanked.

The digital information is clocked from the CPU to the modified 1856 using output sync and ready lines. Sync signals are provided by the 1856 and each sync pulse is counted by a pulse counter as it is sent. The function of the pulse counter is covered when the photopen is discussed.

The scheme used is for the program to build tables in core storage that contain 16 bit words formatted in the following fashion:

BIT	FUNCTION
0	Not used
1-10	Data value
11-12	Selects one of 4 analog to digital converters, 00=X DAC 11= Δ Y DAC 01= Δ X DAC 10=Y DAC
13, 14, 15	Selects one of 8 intensities (000-111)

Each word in the table contains a data value in bits 1 through 10 which will be converted to a voltage by the DACS. Which one of the DACS performs the conversion is determined by bits 11 and 12 of the word, i.e., X, Δ X, Y, or Δ Y. Bits 13, 14 and 15 of the word contain the luminance (intensity) information and are routed to a 5th DAC which provides input for the Z amplifier of the CRT to control the intensity. One of eight intensity levels are available ("000" the beam is completely blanked "111" provides the greatest intensity). The voltage sent to the X amplifier of the CRT is always the sum of the last value converted by the X DAC and the last value converted by the Δ X DAC. Similarly the voltage sent to the Y amplifier of the CRT is always the sum of the last

value converted by the Y DAC and the last value converted by the ΔY DAC.

The following table relates the bits in the 16 bit binary word that are "on" to the analog voltage that will be produced as a result of the bit being turned on. The voltage generated by the DAC is a result of summing the voltage contribution of the bits that are "on".

Data Word Bit	Analog Voltage	
0		
1	2.500	} X, ΔX , Y or ΔY DAC's
2	1.250	
3	.625	
4	.312	
5	.156	
6	.078	
7	.039	
8	.020	
9	.010	
10	.005	
11		
12		
13	1.250	} Z-DAC ONLY
14	.625	
15	.312	

For example suppose the following 16 bit word is sent to the modified 1856 0100100100000011. Because bits 11 and 12 are zero the X DAC will be selected. The X DAC will convert the 10 bit data value to $2.500 + 0.312 + 0.039 = 2.851$ volts. This voltage in turn will be applied to the X amplifier of the CRT. The Z-DAC will convert bits 13, 14, 15 to $0.625 + 0.312 = 0.937$ volts. Because the data value consists of ten bits there a $2^{10} = 1024$ possible voltages that can be produced by the

X DAC and ΔX DAC and also 1024 possible voltages that can be produced by the Y DAC and ΔY DAC. This gives rise to $1024 \times 1024 = 1,048,576$ discrete dots that can be displayed on the screen. The dot size is approximately 30/1000 of an inch so it is possible to overlap dots.

The photopen employed is a Sanders EOPT which is used to trap the position of a particular dot on the display. This is accomplished in the following fashion. The CRT X and Y deflection amplifiers drive the electron beam to the various positions on the phosphor screen which make up a dot display. If the photopen is positioned over a particular dot of the display and activated the change in luminance of that particular dot as the dot is refreshed will trigger the photopen to produce an output signal. This output signal inhibits the 1856 logic from requesting any subsequent data transfers, i.e., no more sync pulses will be sent to the 1800 for the purpose of clocking data to the DACS thus stopping any additional dots from being displayed. Also the output of the photopen is used to provide a process interrupt signal of 1×10^{-6} sec. to interrupt level 11 bit 14. This interrupt is then serviced by an interrupt core load which reads the pulse counter that has counted the number of sync pulses that have been sent to the 1800 to generate the current dot display.

As an example suppose that it is desired to plot 10 points comprising a straight line on the CRT with an intensity of 5. After appropriate scaling the software would build a table of 16 bit words formatted as in Table 1. In writing the software to generate these tables the following convention was adopted. The X and Y absolute commands are used only once, at the beginning of each "picture". They establish the origin of this picture relative to the lower left corner of the CRT screen and thereafter ΔX and ΔY commands are used for all data within the picture.

BITS

WORD #	0	1	10	11,12	13,14,15	
1	X	ABSOLUTE X VALUE	0 0	0 0 0		
2	X	ABSOLUTE Y VALUE	0 1	0 0 0		
3	X	$\Delta X = 0$	1 0	0 0 0		
4	X	$\Delta Y = 0$	1 1	0 0 0		
5	X	ΔX_1	1 0	0 0 0		} DOT ₁
6	X	ΔY_1	1 1	1 0 1		
7	X	ΔX_2	1 0	0 0 0		} DOT ₂
8	X	ΔY_2	1 1	1 0 1		
9	X	ΔX_3	1 0	0 0 0		} DOT ₃
10	X	ΔY_3	1 1	1 0 1		
11	X	ΔX_4	1 0	0 0 0		} DOT ₄
12	X	ΔY_4	1 1	1 0 1		
13	X	ΔX_5	1 0	0 0 0		} DOT ₅
14	X	ΔY_5	1 1	1 0 1		
15	X	ΔX_6	1 0	0 0 0		} DOT ₆
16	X	ΔY_6	1 1	1 0 1		
17	X	ΔX_7	1 0	0 0 0		} DOT ₇
18	X	ΔY_7	1 1	1 0 1		
19	X	ΔX_8	1 0	0 0 0		} DOT ₈
20	X	ΔY_8	1 1	1 0 1		
21	X	ΔX_9	1 0	0 0 0		} DOT ₉
22	X	ΔY_9	1 1	1 0 1		
23	X	ΔX_{10}	1 0	0 0 0		} DOT ₁₀
24	X	ΔY_{10}	1 1	1 0 1		

TABLE 1

This data would then be sent to the modified 1856 using digital/analog output and a data channel by executing an I/O instruction to start the data channel. The data would be clocked from the CPU to the modified 1856 by sync pulses produced by the 1856. Each sync pulse generated will be counted by the pulse counter. Thus referring to the table it would take 24 sync pulses to display the data on the CRT screen one time and the count in the pulse counter would have reached 24 after one display cycle. At the end of this display cycle the pulse counter is reset to zero and the data channel operation is reinitiated again. In order to avoid flicker there should be a minimum of 38 of these display cycles per second. One display cycle is called a refresh.

Now suppose that it is desired to trap, with the photopen, the 4th point in the display. Thus we place the photopen over the 4th point of the display and activate the pen. The change in luminance of the point as it is refreshed will trigger the pen to produce an output signal. This output signal will inhibit the 1856 from sending additional sync pulses to the CPU and consequently no more data will be transferred to the display causing the display to disappear from the screen.

In addition the photopen output signal will cause an interrupt to take place which will start the execution of a core load. One function of the core load is to read the value in the pulse counter which will be 12 for the 4th point. This value is modified by the following algorithm $\left[\frac{\text{Pulse cnt} - 4}{2} \right]^*$ and the modified value is passed through INSKEL COMMON back to the application program and the application programmer can use it to identify the point that was trapped by the photopen. He can then delete, modify or perform other operations on the point as he desires.

* This algorithm is only correct when the table has been formatted for use with the photopen.

1.2 Software Description

1.2.1 Software Concepts

1.2.1.1 Pictures

Conceptually the information displayed on the CRT screen is made up of a number of different "Pictures" currently limited to a maximum of 6. It is a basic property of this set of software that pictures can be handled as independent entities separate and apart from each other, and deleted or included in a display as desired. To develop a picture for the display it is necessary for the application programmer to attach a unique key to the picture called a picture identification (ID). This is done by a call to the subroutine SCALZ. Thereafter, any subroutine call that affects the picture will have this ID as one of its arguments. Once the application programmer assigns an ID to the picture and provides scaling information the process of building and manipulating the picture can begin. For example:

- 1.2.1.1.1 Data can be stored in the picture by calls to PLOTZ, MOVEZ, ALPHZ, INTGZ, and FLPTZ.
- 1.2.1.1.2 The intensity of some or all of the dots in the picture can be altered from the default intensity by calls to INT1Z, INT2Z, INT3Z.
- 1.2.1.1.3 The picture can be relocated on the display screen by a call to RSETZ.
- 1.2.1.1.4 The picture can be expanded or contracted in ordinate or abscissa or both by a call to FACTZ.
- 1.2.1.1.5 The point density of "solid lines" in a picture can be altered by a call to DEN2Z.

- 1.2.1.1.6 Pictures can be stored and retrieved from the 1810 disks through calls to PUTZZ and GETZZ.
- 1.2.1.1.7 Pictures can be included or deleted from a particular display by calls to ONZZ and OFFZ.
- 1.2.1.1.8 The core storage occupied by a particular picture when the picture is no longer required can be freed by a call to FPICZ or a picture can just be cleared of information by a call to CPICZ or RSCLZ.

Thus we have a number of independent entities called pictures which can be operated on by a general set of subroutines. After construction of the pictures is complete, those pictures turned on are displayed by a call to EXCRT.

1.2.1.2 Dynamic Pictures

Another useful property of this software set is the ability to change or move a picture after each refresh of the display. Because the screen phosphor is of short persistence the display must be refreshed about 35 to 40 times per second to avoid the sensation of flicker. The advantage of the short persistence phosphor is the fact that the display can be changed after each refresh without leaving an image of the old display for a visually detectable time. The software is implemented such that a call to EXCRT is required before any pictures are actually displayed on the CRT screen. That is, all pictures would be constructed and turned on and then a call would be made to EXCRT to actually display the pictures on the CRT screen. EXCRT provides for two options. The first option, CALL EXCRT (0), will display

the pictures that are turned on until a photopen interrupt takes place, i.e., continual refresh of the picture takes place. The second option, CALL EXCRT (1), will return control to the next instruction following the call after each refresh of the display. Thus with option 2 the user can change the position or contents of pictures after each refresh and thus give a dynamic characteristic to the display.

For example it is possible to display a ball bouncing on a flat surface by storing the dots representing the ball in a particular picture and changing the origin of this picture after each refresh and storing the dots representing the surface in another picture which does not move. The amount that the ball picture is moved in ordinate and abscissa after each refresh can be computed using the laws of dynamics of a body in a gravitational field striking a surface with some predefined coefficient of restitution. A realistic model of the actual physical situation can thus be produced.

1.2.1.3 Function Matrix and Photopen

Available to the application programmer is the ability to include in the display a picture which consists of the "Function Matrix" (FM). This is a one inch square picture which has four rows of hexadecimal characters formatted as below:

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

The inclusion of the FM in a display gives the programmer the ability to make decisions in his program using the photopen.

By positioning the photopen over one of these characters and activating the pen the hexadecimal character value (in integer format) trapped by the pen is returned through INSKEL COMMON to the application program.

The application program can make a decision based on the value returned. For example an integer value of 10 corresponding to the "A" in the FM might indicate to the program that it should expand the ordinate of a certain picture by a fixed amount or an integer 11 corresponding to the "B" in the FM might indicate to the program that it should delete from the display a certain picture. If the photopen is used to trap a dot outside of the FM, i.e., a dot in another picture, the integer value of 16 is returned along with the index of the dot trapped and the ID of the picture which contained the trapped dot. This allows the application program to operate on this data. For example, it might be desirable to delete the data from the picture for some reason. Routines are provided to delete this data from the picture and from the user data arrays.

1.2.2 Software Definitions

Certain definitions necessary for the use of this equipment and its associated software are given below.

1.2.2.1 Display - This consists of all the information displayed on the CRT screen under computer control. This information is presented as a pattern of dots of variable intensity arbitrarily positioned on the screen. The usable display area is approximately seven by nine inches. The actual screen size is 8 by 10 inches.

1.2.2.2 Picture - A logical grouping of data for display on the CRT screen. Under current implementation up to six pictures may be in use and displayed simultaneously. Each picture is a discrete entity and may be modified by the user, moved about the display area, turned on or off, or stored on the disk for future reference without affecting the rest of the display.

1.2.2.3 Origin - The lower left corner of the display area or of an individual picture. No picture origin may be placed below or to the left of the display area origin. No information in an individual picture may appear below or to the left of that picture's origin.

1.2.2.4 Raster unit - The smallest increment between two dots of the display. There are one hundred raster units per inch.

1.2.2.5 Picture ID - Each picture has a unique ID assigned by the user, which is a variable in the calling program. This variable (referenced as ID in the following subroutine descriptions) functions as a key and is used to identify the picture which a subroutine call is to affect. The picture ID must not be used

for any other purpose or modified in any way. The software inserts an address into this name which points to a unique Picture Pointer Table (PPT).

- 1.2.2.6 Intensity - The brightness of individual dots or portions of pictures may be specified by the user, using a scale of 1 (dimpest) to 7 (brightest). The standard (unspecified) intensity is five.
- 1.2.2.7 Density - A "solid" line in the display consists of a series of consecutive dots. The spacing of these dots may be specified by the user. The density is given in raster units and has a standard (unspecified) value of five (i.e., twenty dots per linear inch).
- 1.2.2.8 Picture Pointer Table (PPT) - A data list (internally generated by the software) containing all pointers and control information necessary for the generation, manipulation and display of a picture.
- 1.2.2.9 Data Table - A block of CRT coded information representing user data. All information to be displayed is translated to CRT code and entered into a data table. All data tables for a given picture are chained together. The first and last table in the chain are linked to the PPT for that picture.
- 1.2.2.10 Free Table List (FTL) - A list of all unassigned data tables. When needed, an empty table from this list is linked to the PPT of a picture requesting more space.
- 1.2.2.11 Free Disk Area (FDA) - An element of the CRT 1810 Disk Process Working Storage named CRTZZ. This element is used by PUTZZ/GETZZ for temporary picture storage. This allows a picture's data tables to

be freed for use by another picture without destroying the data. Only one picture may reside in a given Disk Area (DA). The number of 1810 sectors in each element is determined by the length of the Free Table List. The number of FDA's may not exceed the maximum allowable number of pictures, and may even be less. This number is a function of the available 1810 disk space, not a function of the CRT software.

1.2.2.12 Data Table Area - An integer array defined in the main program, which contains the data tables whose contents are transferred to the digital to analog converters at display time. These tables are generated by the software described in this document. The dimension of this array should be a multiple of 323, and under the current implementation can have a maximum dimension of 5168. Once a program has been made operational, the dimension of this array may be optimized to save core by analysis of a dump of the CRT Communications Area. This dump is obtained by use of the subroutine COMDP.

1.2.2.13 INSKEL COMMON - INSKEL COMMON is a labelled common area set aside in the skeleton. Its length, which is specified during system generation, is 150 in the current implementation, of which the last nine words are reserved for communication between the CRT routines. The values assigned to these words are shown in Table 2.

In order to reference these words the following statement must be included in a FORTRAN program:

```
COMMON/INSKEL/IFILL(141), LIND, LAD,LIN,LPPT,LR,LY,LX,LID,LFP
```

The appropriate statements for an assembler language routine,

directly following the //ASM card are:

*ONE WORD INTEGERS

*COMMON/INSKEL/IFILL(141),LIND,LAD,LIN,LPPT,LR,LY,LX,LID,LFP

Inskel Common Table

<u>Word</u>	<u>Name</u>	<u>Content</u>
150	LFP	Address of 1st DAO table being displayed.
149	LID	The ID or name of the picture in which the last photopen interrupt occurred.
148	LX	The X coordinate of the point at which the last photopen interrupt occurred, expressed in raster units relative to the display area origin.
147	LY	The Y coordinate of the point at which the last photopen interrupt occurred, expressed in raster units relative to the display area origin.
146	LR	This variable conveys a value selected with the photopen to the user's program if he has included the function matrix in his display (see subroutine LIPNP.) If the user selects one of the characters within the matrix, the corresponding numeric value between 0 and 15 will be returned in LR. If an interrupt is caused at a point outside of the function matrix the value of LR will be 16. If no interrupt has occurred, LR is set to -1.
145	LPPT	The ID or name (PPT address) of the picture in which a user response type error occurred.
144	LIN	The pulse count within the individual picture for the point at which an interrupt occurred.
143	LAD	Address of the on list (ONLST)
142	LIND	After a call to DLETZ, INT3Z or INDXZ LIND contains the Fortran index of the point at which the interrupt was caused, if this is applicable (see <u>Indexed Data Arrays</u> and the subroutine descriptions.)
1-141		Not used by the CRT routines.

TABLE 2

1.2.3 Subroutines Directly Available to the Application Programmer

1.2.3.1 ALPHZ CALL ALPHZ(ID,X,Y,ISIZ,N,IARRY)

Subroutine ALPHZ causes a set of specified alphanumeric or special characters to be displayed in the picture with a key of ID. The character plotting will begin at a point (X,Y) inches from the origin of the picture. X and Y must be nonnegative. The integer ISIZ specifies the character size in 0.05 inch units (i.e., ISIZ of three will specify a set of characters 0.15 inch height). N is the number of characters to be displayed. IARRY is the name of an integer array whose first N values specify the characters to be displayed according to the codes given in Table 3.

TABLE 3

CHARACTER	CODE	CHARACTER	CODE
0	0	N	25
.	.	O	26
°	°	P	27
•	•	Q	28
9	9	R	29
-	10	S	30
+	11	T	31
A	12	U	32
B	13	V	33
C	14	W	34
D	15	X	35
E	16	Y	36
F	17	Z	37
G	18	period	50
H	19	blank	51
I	20	(38
J	21)	39
K	22	=	40
L	23	X	41
M	24	/	42

1.2.3.2 CPICZ CALL CPICZ (ID)

Clear a picture. This routine removes all data from the picture and reinitializes its PPT. One data table is retained (any 'active' picture must have at least one table assigned). If the picture is inactive, the disk area is freed and a new data table is assigned. The origin, scaling factors and picture on/off condition are unchanged. This allows the user to remove data from a picture without freeing and then redefining the picture.

1.2.3.3 DEN2Z CALL DEN2Z (ID,N)

Subroutine DEN2Z causes all request for solid line displays in the picture ID to have density N, where N is an integer between one (1) and thirty-one (31). Only requests made subsequent to this subroutine call are affected.

1.2.3.4 DLARY CALL DLARY(LEN,N,X,Y,...)

Subroutine DLARY removes the value corresponding to a specified index from each of an arbitrary number of floating point data arrays and compresses the arrays to fill the vacancies. This routine will normally be used in conjunction with DLETZ. LEN is the current length of each of the N arrays X,Y, etc. The element corresponding to index LIND (word 142 of INSKEL COMMON) will be removed from each of the arrays, and LEN will be decremented by one by the subroutine. If LIND is zero, no action will be taken.

For example, suppose that in a program the following is true.

```
LEN = 3    X(1)= 1.0    X(2)= 2.0    X(3)= 3.0
           Y(1)= 2.4    Y(2)= 8.9    Y(3)= 3.5
```

If the following coding

```
LIND=2  
CALL DLARY(LEN,2,X,Y)
```

were executed, the results

would be;

```
LEN=2   X(1)= 1.0   X(2)= 3.0  
        Y(1)= 2.4   Y(2)= 3.5
```

Note that LEN must be an integer variable name (i.e., unlike many Fortran calls the integer itself may not be used in the calling statement) and that the program must include the appropriate INSKEL COMMON statement.

1.2.3.5 DLETZ

CALL DLETZ

Subroutine DLETZ allows deletion from the display of dot selected by the photopen. If DLETZ is called immediately subsequent to the photopen interrupt, the selected dot is brightened to maximum intensity (seven) and the display is turned on again. This allows the user to ensure that the proper dot was identified. He must now indicate to the program by use of the photopen and function matrix (see LIPNP) whether he wishes the brightened dot to be removed. If he indicates any value from 0 to 7 in the matrix (the upper half) the dot will be removed from the display and LIND (word 142 of INSKEL COMMON) will be set to the index appropriate to this point. If a value between 8 and F (the lower half of the function matrix) is selected, the dot will not be deleted from the display, but will be reset to its original brightness, and LIND will be set to zero.

1.2.3.6 EXCRT CALL EXCRT (I)

This routine causes all pictures that have been turned "on" by the ONZZ subroutine to be displayed on the Cathode Ray Tube. If I=0 the display will be continually refreshed at the maximum rate consistent with the quantity of data being displayed and the rate at which data can be transferred to the display. If I=1 control will be returned to the next sequential instruction after the call to EXCRT after each refresh.

1.2.3.7 FACTZ CALL FACTZ (ID,XF,YF)

FACTZ causes the entire picture ID to be expanded or contracted in the X direction relative to the origin by scaling factor XF, and in the Y direction by YF. All subsequent additions to the display will be multiplied by these factors. Calls to FACTZ are cumulative; i.e., two successive calls to FACTZ with XF=0.5 will cause all X dimensions to be shrunk to 0.25 their original size. Similarly, if a call with XF=0.5 has been made, a call with XF=2.0 would be necessary to restore the picture to its original dimensions. The origin remains unchanged by this call.

1.2.3.8 FLPTZ CALL FLPTZ(ID,X,Y,ISIZ,F)

Subroutine FLPTZ causes the floating point number F to be displayed at a point (X,Y) inches from the origin of picture ID. X and Y must be nonnegative. Five significant digits of the number F will be presented in an appropriate form selected by the subroutine. ISIZ has the same significance as in subroutine ALPHZ.

1.2.3.9 FPICZ CALL FPICZ (ID)

This routine frees a picture and suspends definition of a picture. All assigned tables are returned to the FTL. The PPT status area is cleared and the characteristic area is reset. If the picture was on, it is turned off. If it was inactive, the disk area is freed. The user picture name (ID) is cleared to zero. Note, it is not necessary to call CPICZ first.

1.2.3.10 GETZZ CALL GETZZ(ID)

'Get' the picture from disk. Data tables are removed from the FTL and the picture is copied serially from the DA to core, one sector per data table. If there are not enough data tables available, a user response error will occur. The picture will not be turned on. The status of the PPT is set to make the picture active. The DA is freed.

1.2.3.11 GTXYZ CALL GTXYZ (ID,IX,IY)

Subroutine GTXYZ returns the current coordinates of the picture ID's origin in variables IX and IY. IX and IY are specified in raster units.

1.2.3.12 INDXZ CALL INDXZ

A call to INDXZ subsequent to a light pen interrupt will cause the index of the selected display point to be set in LIND (word 142 of INSKEL COMMON).

1.2.3.13 INITZ CALL INITZ(IAREA,LEN,ERROR)

This routine initialize the CRT system. The primary function of this routine is to fragment the work space IAREA provided by the user into 323 word tables, count these tables (NFTL) and assign them to the FREE TABLE LIST, FTL. Each table is formatted

as it is assigned to the list. LEN is the dimension size of IAREA and determines the number of tables allocated. The user error recovery routine ~~ERROR~~ is also recognized and saved for future use. Inclusion of a user written error routine is optional, however, if not included the word "ERROR" must appear in call sequence.

1.2.3.14 INTGZ CALL INTGZ(ID,X,Y,ISIZ,I)

Subroutine INTGZ causes the integer I to be displayed at a point (X,Y) inches from the origin of picture ID. X and Y must be nonnegative. ISIZ has the same significance as in subroutine ALPHZ.

1.2.3.15 INT1Z CALL INT1Z(ID,N)

A call to subroutine INT1Z causes picture ID to be made brighter or darker according to the value of N specified. All dots of the picture, regardless of their previous intensity, are given intensity N, where N is an integer between one and seven. All dots of picture ID created subsequent to this call will have intensity N.

1.2.3.16 INT2Z CALL INT2Z(ID,N)

Subroutine INT2Z causes all dots of picture ID created subsequent to this call to have brightness N, where N is an integer between one and seven.

1.2.3.17 INT3Z CALL INT3Z(N)

Subroutine INT3Z is used in conjunction with the photopen interrupt feature. If this subroutine is called immediately subsequent to a photopen interrupt the individual dot which was selected with the photopen will have its intensity changed to N, where N is an integer between one and seven. LIND (word 142 of INSKEL COMMON) will be set to the index appropriate to

this point if it is part of an indexed array.

1.2.3.18 LIPNP CALL LIPNP(ID)

A call to LIPNP causes the function matrix (a 4x4 array containing the 16 hexadecimal digits 0 - F) to be displayed in the upper right hand corner of the display screen. The matrix is stored in picture ID, which must not be used for any other display. Unlike all other pictures used, no call to SCALZ for picture initialization is necessary, and no call to ONZZ is necessary to cause the matrix to be displayed. Subroutines RSETZ or INT1Z may be used respectively to change the position of the matrix or to modify its intensity. The use of the matrix will be discussed in the section on interrupt (photopen) programming (see Section 1.2.5.1).

1.2.3.19 MOVEZ CALL MOVEZ(ID,X,Y,N)

Subroutine MOVEZ causes the CRT electron beam to be moved to a position (X,Y) inches from the origin of picture ID. If N equals one only a dot at (X,Y) will be displayed; if N is equal to zero a line will be traced from the last dot displayed in picture ID. If this is the first display command issued for picture ID this last dot will be the picture origin.

1.2.3.20 OFFZ CALL OFFZ(ID)

Turn the picture 'off'. The picture is removed from the display. The picture name is removed from the on list (ONLST) and made unavailable for display.

1.2.3.21 ONZZ CALL ONZZ(ID)

Turn the picture 'on'. The picture resident in core is added to the display. Functionally, the picture name is entered in the on list (ONLST) of pictures available for display.

1.2.3.22 PLOTZ CALL PLOTZ(ID,X,Y,N)

Subroutine PLOTZ causes the CRT electron beam to be moved to the position (X,Y) relative to the picture origin of picture ID, where X and Y are specified in user units. If N equals one only a dot (X,Y) will be displayed; if equal to zero a line will be traced from the last dot displayed in picture ID. If this is the first display command issued for picture ID this last dot will be considered to be the picture origin.

1.2.3.23 PUTZZ CALL PUTZZ(ID)

This routine 'Puts' the picture on disk. All data tables in the chain linked to this picture's PPT are serially copied to a Free Disk Area of the file CRTZZ in 1810 disk process working storage. The picture is stored one data table per sector. All data tables are freed and returned to the Free Table List (FTL) for reuse. If the picture is on, it is turned off. The status of the PPT is set to make the picture inactive.

1.2.3.24 RSETZ CALL RSETZ(ID,IX,IY)

Subroutine RSETZ resets the origin of the picture ID to the position (IX,IY) relative to the display area origin. The entire picture is moved to the new location. IX and IY are specified in raster units.

1.2.3.25 RSCLZ CALL RSCLZ(ID,XL,XU,XS,YL,YU,YS)

Subroutine RSCLZ clears the current contents of picture ID, and rescales according to the latter six parameters. These parameters have the same significance as in subroutine SCALZ. The expansion factors (see FACTZ) are set to 1.0 for both X and Y. Other characteristics of the picture such as position or the origin,

intensity, density and on or off condition are left unchanged.

1.2.3.26 SCALZ CALL SCALZ(ID,IX,IY,XL,XU,XSIZ,YL,YU,YSIZ)

A call to SCALZ is necessary to initialize the picture ID, and must be made before any information is displayed in the picture.

IX and IY give the position of the origin of picture ID in raster units relative to the origin (lower left corner) of the display screen. This position of the origin may later be modified by use of the subroutine RSETZ.

XL and YL are the X and Y coordinates of the origin in user units. These are the lower bounds of X and Y data which may be displayed in picture ID. XU is the value in user units of X at XSIZ inches from the origin. YU is the value in user units of Y at YSIZ inches from the picture origin.

1.2.4 System Subroutines

1.2.4.1 COMDP CALL COMDP

A call to this subroutine causes a dump of the CRT communication area on the 1443 printer. This dump includes the number of free areas, the list of free areas, and the picture pointer tables.

1.2.4.2 DPICZ(ID) - This subroutine defines a picture. This routine assigns a PPT to the user picture name and initializes the PPT by assigning the first data table to the PPT. This routine should not be called directly by the user. To provide appropriate scaling information, the user should call SCALZ (which in turn calls DPICZ).

1.2.4.3 ERRZ(I) - Write an error message to the user. The program disposition is determined by the value of I. Certain error numbers are warnings only, some are terminal errors, and a few require a user response. See the list of error messages in Appendix A. This routine may be called only by the CRT software not by the application program.

1.2.4.4 FDSAZ(DA) - Free a disk area. An element of CRTZZ is freed by returning the first disk sector address to the FDA table. The DA is contained in the accumulator. This routine may be called only by the CRT software.

1.2.4.5 FMTZ(ADDR) - This routine formats the data tables for use by digital/analog output. The address of the data table is in the accumulator. This routine may be called only by the CRT software.

1.2.4.6 GTABZ(ADDR) - Fetch a free data table from the FTL. This routine may be called only by the CRT software.

1.2.4.7 LITPN

This program is a core load that operates in partition

#1 under MPX and responds to the photopen interrupt
(level 11 bit 14). It communicates with the user programs through
INSKEL COMMON in the following fashion:

TYPE	INSKEL COMMON WORD #	
I	150	Address of 1st DAO table being displayed
I	149	Picture ID containing point trapped
I	148	X coordinate in rasters of point trapped
I	147	Y coordinate in rasters of point trapped
I	146	Return code
I	145	(Not set by this subroutine)
I	144	Value read from pulse counter

The return codes have the following definitions:

A value in the range zero through 15 is returned when the same
value is trapped with the photopen from the function matrix.
Sixteen is returned if a point other than the function matrix
is trapped.

After a photopen interrupt is serviced by the LITPN core
load control is returned to the next sequential instruction.

1.2.5 Examples

1.2.5.1 An Example of Data Editing Via Interactive Software

The program LSQRF (See Figure 1a, b, c)

illustrates data editing and deletion techniques. This program reads a set of X and Y data from cards and displays the data on the CRT screen. A linear least squares fit to the data is performed and the results of the fit are displayed, both in the form of a straight line of contrasting intensity superimposed on the data, and as a pair of numerical coefficients for the equation of the fitted line. The user may use the photopen to delete undesired points from this data set, upon which a new least squares fit will immediately be made and displayed. The user may also have the current data set and fit coefficients written on the 1443 line printer at any time and may regenerate the entire original data set if he wishes.

The first statement of this program, EXTERNAL ERROR, is mandatory in all programs. It refers to an optional user written error handling routine ERROR (see Section 1.2.5.2). If the user does not choose to write such a routine a dummy (non-functional) subroutine of this name which is stored on disk will be added to the core load.

The dimension statement contains arrays for the X and Y data, a data table area IA of 3230 words, an array IALF which will contain the characters for the alphanumeric legend, and two input arrays. The COMMON/INSKEL/ statement allows the program to use the named variables to communicate with the photopen interrupt routine.

The data table area and CRT subroutines are initialized by the mandatory call to INITZ. Immediately following it (as it must) is a call to LIPNP which causes the function matrix to be generated in picture ID1 and added to the display. Note that unlike all other pictures in the display, no call to either SCALZ or ONZZ for ID1 is necessary, since LIPNP performs the functions of these calls automatically.

The call to RSETZ moves the function matrix to a point in the display which is more convenient for this program. This point is one inch to the right of, and 5 inches above the display origin. A call to SCALZ then initializes picture ID3 which will contain certain non-changing portions of the display. A call to DEN2Z specifies that all "solid" lines drawn in ID3 will consist of dots at ten raster unit intervals (i.e., 10 dots per inch). The call to MOVEZ then causes a base line of this density to be drawn from the picture origin to a point six inches to its right. Three successive calls to ALPHZ cause the messages "Y=AX+B", "A= " and "B= " to be added to the lower right corner of the picture. Finally a call to ONZZ turns picture ID3 "on", that is, causes it to be added to the active display.

Data is then read from cards into arrays ZX and ZY until a blank card is encountered. When this occurs a branch is made to statement 30 where LEN is established as the data count (number of points). A call to SCALZ initializes picture ID2 which will contain the data. The origin of this picture (like that of ID3) is set one inch above and to the right of the display origin.

A six by four inch area for the data is defined, the lower and upper bounds for the X data of 250 and 700, and Y bounds of 0 and 16. The data from arrays ZX and XY is stored into X and Y, which will contain the current (i.e., edited) data set.

At statement 31 a call to INT2Z sets the intensity of the picture to 5 for future additions to the picture. This statement has no practical effect the first time it is executed since the default or original intensity of all pictures is set at 5. A loop using PLOTZ then calls for all the data in arrays X and Y to be included in picture ID2. Note that since the Fortran indexing feature of the photopen interrupt routines is to be used, these data points are the first information stored in this picture. A call to LSQR then performs a least squares fit to the data, and returns the fit coefficients, A and B. Two calls to FLPTZ cause these numbers to be formatted and displayed in the proper place on the CRT screen. The intensity (for subsequent display information) is now lowered to 2. The next calls to PLOTZ cause the line representing the least squares fit to be drawn in the data area at this lower intensity. This difference in intensity between the fit line and the data points provides contrast to prevent confusion and allows the user to "tune out" the line when necessary by operating the display unit intensity control knob. This is necessary where a data point to be identified with the photopen lies so close to the fit line that distinguishing the proper point would be difficult. Finally a call to ONZZ adds picture ID2 to the active display, and the display is complete.

At statement 52 a call to EXCRT activates the display which will continue to be refreshed until a photopen interrupt occurs. The value returned in INSKEL COMMON variable LR when an interrupt occurs determines what action will be taken by the program.

If the interrupt was caused by the photopen trapping a data point, then LR equals 16 and a branch is made to statement 3. Here a call to DLETZ causes the display to be turned on again with the trapped point marked by being displayed with maximum intensity (seven). The user must now indicate whether he wishes the marked point to be deleted or not. If not he traps any character in the lower half of the function matrix with the photopen and DLETZ returns with INSKEL COMMON variable LIND set to zero. DLARY takes no action due to this zero value, but an error message is printed on the typewriter.

If, on the other hand, the user traps any character in the upper half of the function matrix this indicates to DLETZ that the selected point is to be deleted. The FORTRAN index of the indicated point is returned in LIND, and routine DLARY deletes the coordinates of this point from arrays X and Y, packs the arrays, and decrements LEN by one.

Whether the user chose to delete the point or not, the call to RSCLZ then clears and reinitializes picture ID2. A transfer is made to statement 31 where the picture intensity is reset to five, the current set of data points are added to ID2, and a new fit is calculated. When the creation of the new display is completed, the display is reactivated at statement 52.

If, during this display, the photopen is used to trap any character 0 through B of the function matrix, a value of one, two or three is computed for IND and a branch is made to statement 1. Here also, a call to RSCLZ causes picture ID2 to be cleared and reinitialized. LEN is set to its original value, and a branch to statement 34 causes the entire original data set to be regenerated.

If the photopen is used to trap any character in the lowest line (characters C-F) of the function matrix, a branch is made to statement 2 where the current contents of the X and Y arrays are printed out on the 1443 line printer along with the corresponding values of A and B. A branch to statement 52 then causes the display to be reactivated.

```

EXTERNAL ERROR
C THIS PROGRAM READS A SET OF CARDS IN F6.2,F6.1 FORMAT CONTAINING
C SORT(TEMP) FOLLOWED BY VELOCITY, FOR VARIOUS SPACECRAFT, AND DISPLAYS
C THE LEAST SQUARES FIT ON THE CRT. DATA EDITING AND PRINTOUT OF RESULTS
C CAN BE DONE. 3/72 T.P. CARLETON
  DIMENSION X(100),Y(100),IA(3230),IALF(25),ZX(100),ZY(100)
  DATA IALF/36,51,40,51,12,35,11,13,12,51,40,51,13,51,40,51/
  COMMON/INSKEL/ IFILL(141),LIND,LAD,LIN,LPPT,LR,LLY,LX,LID,LFP
  CALL INITZ(IA,3230,ERROR)
C GENERATE FUNCTION MATRIX AND MOVE IT
  CALL LIPNP(ID1)
  CALL RSETZ(ID1,100,500)
  CALL SCALZ(ID3,100,100,250.,700.,6.,0.,16.,4.)
C DRAW BASE LINE (10 PTS./IN.) AND LEGEND IN PICTURE ID3
  CALL DEN2Z(ID3,10)
  CALL MOVEZ(ID3,6.0,0.0,0)
  CALL ALPHZ(ID3,4.0,1.0,3,8,IALF)
  CALL ALPHZ(ID3,4.0,0.67,3,4,IALF(9))
  CALL ALPHZ(ID3,4.0,0.33,3,4,IALF(13))
C ADD ID3 TO THE DISPLAY
  CALL ONZZ(ID3)
  N=1
C READ IN DATA AND INITIALIZE PICTURE ID2
33 READ(2,100)ZY(N),ZX(N)
100 FORMAT(F6.2,F6.1)
  IF(ZY(N))30,30,32
32 N=N+1
  GO TO 33
30 LEN=N-1
  CALL SCALZ(ID2,100,100,250.,700.,6.,0.,16.,4.)
34 DO 51 I=1,LEN
  X(I)=ZX(I)
51 Y(I)=ZY(I)
C PLOT DATA IN ID2
31 CALL INT2Z(ID2,5)
  DO 50 I=1,LEN
50 CALL PLOTZ(ID2,X(I),Y(I),1)
C DO LEAST SQUARES FIT AND ADD NUMERICAL PARAMETERS TO ID2
  CALL LSQR(X,Y,LEN,A,B)
  CALL FLPTZ(ID2,4.6,0.67,3,A)
  CALL FLPTZ(ID2,4.6,0.33,3,B)
C LOWER INTENSITY AND DRAW LEAST SQUARES FIT LINE THROUGH DATA
  CALL INT2Z(ID2,2)
  YLSQ=A*X(1)+B
  CALL PLOTZ(ID2,X(1),YLSQ,1)
  YLSQ=A*X(LEN)+B
  CALL PLOTZ(ID2,X(LEN),YLSQ,0)
  CALL ONZZ(ID2)
C DISPLAY DATA ON CRT UNIT
52 CALL EXCRT(0)
  IND=LR/4+1
C BRANCH TO INTERRUPT PROCESSING
  GO TO (1,1,1,2,3),IND
C DELETE DATA POINT INDICATED BY LIGHT PEN
3 CALL DLETZ
  CALL DLARY(LEN,2,X,Y)
  CALL RSCLZ(ID2,250.,700.,6.0,0.,16.,4.0)
  GO TO 31
1 CALL RSCLZ(ID2,250.,700.,6.0,0.,16.,4.0)
  LEN=N-1
  GO TO 34
2 WRITE(3,101)(X(I),I=1,LEN)
101 FORMAT('0',20F6.1)
  WRITE(3,102)(Y(I),I=1,LEN)
102 FORMAT(' ',20F6.2)
  WRITE(3,103)A,B
103 FORMAT('0 A = ',E11.4,', B = ',E11.4)
  GO TO 52
99 CALL EXIT
END

```

FIGURE 1a

```
      SUBROUTINE LSQR(X,Y,N,A,B)
      DIMENSION X(1),Y(1)
      IF(N-1)8,8,2
2     SX=0
      SX2=0
      SXY=0
      SY=0
      DO 1 I=1,N
      SX=SX+X(I)
      SX2=SX2+X(I)*X(I)
      SXY=SXY+X(I)*Y(I)
1     SY=SY+Y(I)
      DIV=SX*SX-N*SX2
      A=(SX*SY-N*SXY)/DIV
      B=(SX*SXY-SX2*SY)/DIV
      GO TO 9
8     A=0.0
      B=0.0
      WRITE(3,100)
100  FORMAT('0INSUFFICIENT POINTS FOR FIT')
9     RETURN
      END
```

FIGURE 1b

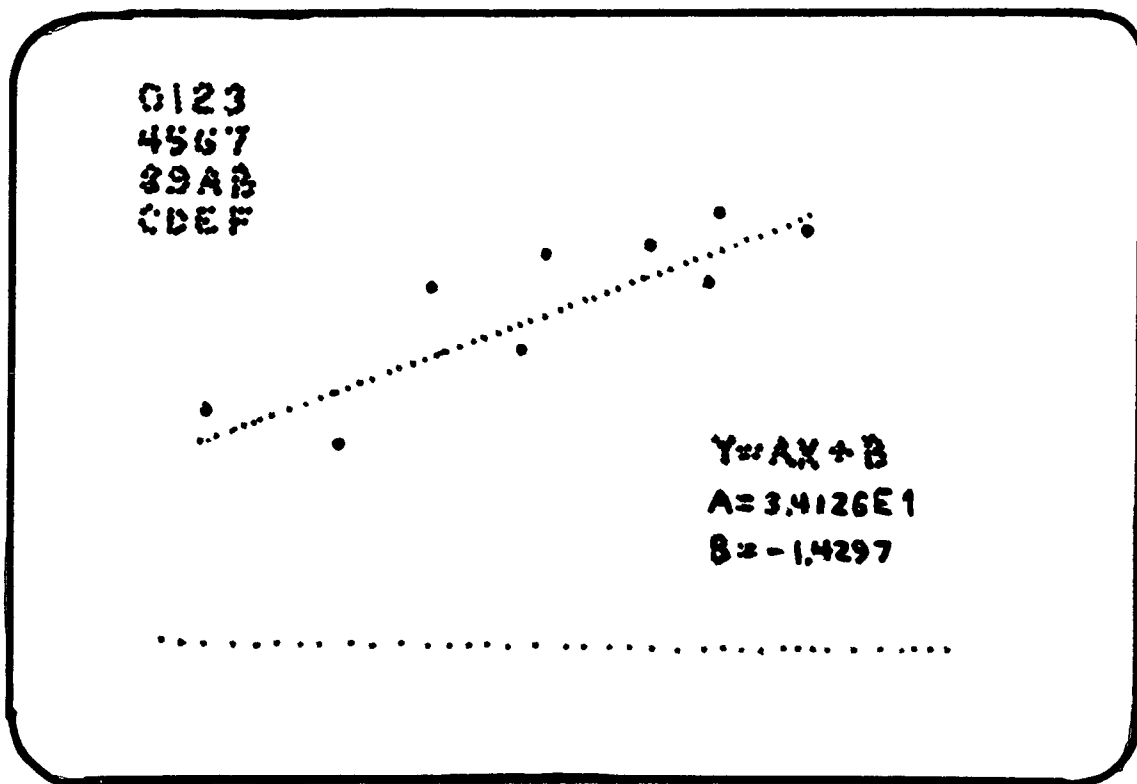


FIGURE 1c

1.2.5.2 An Example of the User Error Recovery Option

When the interactive graphics software detects an error condition, there are three possible disposition states: warning, termination, and user response (see Appendix A). The user response option is usually related to space problems, e.g., space in core, or space on disk. The user has the option of responding with some action which will free the appropriate type of space, or terminate the job. If space is freed, the job will continue from the point where the error took place. The user should be aware that during the process of correcting one error condition, should any additional error condition arise (even a warning level) the job will fail.

Under certain conditions, the user may have greater control in processing a user error if the user treated the error routine as the 'main' program. That is, make the error routine serve the dual function of picture generation and error recovery. Figure 2 is an example of such a case. In the main program, the user allocates table space and assigns subroutine CRT as the error processing routine in statement 1. The user then calls subroutine CRT with a negative argument (a non-existent error code), in statement 2. It should be noted that the user cannot return to the main program once a user error has occurred. Subroutine CRT first checks the error code in statement 10. The negative code indicates the entry from the main program, so the branch is to statement 20, where the function matrix (ID0), the data picture (ID1), and grid picture (ID3) are initialized.

A branch is taken to statement 60 where the data picture ID1 is generated. The picture is a sine wave, but of far greater size than can be contained in the data tables allocated. When all space is used, a branch is taken to ERRZ, where a message is written to the user indicating a Z03 TABLES error has occurred. This is a user response error, and ERRZ will call CRT with an argument of 3. When CRT is entered, the branch at statement 10 is taken to statement 50, where, after determining that the error was in fact number 3, the picture is displayed with as much data as could be contained in it, statement 51. Then, using the photopen, the user may designate one of three responses: 1) go to statement 90 and terminate the job, 2) go to statement 95 and put picture ID3 (grid) on disk, making its data table available for use by ID1, 3) go to statement 56 and clear ID1, rescaling the picture to begin where the previous picture left off. If ID3, the grid picture, is put on disk, the return from CRT is via ERRZ to the CALL PLOTZ at statement 70 that initially generated the error and processing continues normally from that point. If the picture ID1 is cleared and rescaled, processing will continue from statement 60.


```
// JOB      X      X      X      28 MAR 72 14.349 HRS
// FOR MAIN  2= MAR 72 14.349 HRS
*IOCS(1443 PRINTER)
*LIST ALL
  EXTERNAL CRT
  DIMENSION IA(1293)
  N=1293
  1  CALL INITZ(IA,N,CRT)
    I=-1
  2  CALL CRT(I)
    CALL EXIT
  END
MAIN
DMP FUNCTION COMPLETED
// FOR CRT   28 MAR 72 14.354 HRS
*LIST ALL
C
C      PROGRAM TO DEMONSTRATE USE OF USER ERROR RECOVERY SUBROUTINE
C
  SUBROUTINE CRT(I)
  COMMON/INSKEL/ IFILL(144),IPPT,ICODE
  DATA XMIN/0.0/,IHY/250/,W/3.14/
  10  IF(I) 20,20,50
C
C      GO HERE ONLY ON ENTRY FROM 'MAIN PROGRAM'
  20  CONTINUE
    CALL LIPNP(ID0)
    CALL RSETZ(ID0,600,250)
    CALL SCALZ(ID1,0, IHY ,0.,    W,4.,-1.,1.,2.)
    CALL ONZZ(ID1)
    CALL SCALZ(ID3,0, IHY ,0.,    W,4.,-1.,1.,2.)
    CALL ONZZ(ID3)
    CALL PLOTZ(ID3,4.0,0.,1)
    CALL PLOTZ(ID3,0.,0.,0)
    MIN=1
    MAX=2000
    GO TO 60
C
C      NORMAL ENTRY FROM ERRZ
  50  CONTINUE
    WRITE(3,150) I,IPPT
    IF(I-3) 99,52,99
  52  CONTINUE
    WRITE(3,160) MIN,K
  51  CALL EXCRT(0)
    ICODE=ICODE+1
    GO TO (56,56,56,56,95,95,95,95,93,93,93,93,90,90,90,90,90),ICODE
C
C      CLEAR PICTURE AND RESET SCALE
  56  CONTINUE
    MIN=K
    XMIN=MIN/50.
    XMAX=XMIN+W
    CALL RSCLZ(ID1,XMIN,XMAX,4.0,-1.0,1.0,2.0)
C
C      GENERATE DATA PLOT
  60  CONTINUE
    CALL PLOTZ(ID1,XMIN,Y,1)
    DO 75 K=MIN,MAX
      X=K/50.0
      Y=SIN(W*X)
  70  CALL PLOTZ(ID1,X,Y,0)
  75  CONTINUE
    CALL EXCRT(0)
C
C      NORMAL EXIT - NO RETURN TO 'MAIN PROGRAM'
  90  CONTINUE
    WRITE(3,170) K
    CALL EXIT
C
C      RETURN GRID TO CORE AND CONTINUE PROCESSING
  93  CONTINUE
    CALL CPICZ(ID1)
    CALL GETZZ(ID3)
    CALL ONZZ(ID3)
    GO TO 56
C
C      PUT GRID ON DISK FREEING ITS SPACE
  95  CONTINUE
    CALL PUTZZ(ID3)
C
C      RETURN FROM ERROR FIX + CONTINUE PROCESSING
  99  CONTINUE
    RETURN
  150  FORMAT(' USER ERROR NO. 2',I2,' IN PICTURE ID ',I7)
  160  FORMAT(45X,' PLOT COMPLETED FROM ',I4,' TO ',I4)
  170  FORMAT(/,' *** CRT TERMINATED AT K=',I4,' ***')
  END
```

FIGURE 2

2.0 Software Internals

2.1 Description of Tables

Most of the information the CRT system uses to control its operation is maintained in a block of tables called the CRT Communications Area (CCA). The organization of CCA is controlled during assembly by two equates, NPICT and NTABL. See Figure 3 for a map of CCA. NPICT is the maximum number of pictures that will be allowed and NTABL is the maximum number of data tables allowed. The elements of CCA are described below.

USERZ - Address of the user error recovery subroutine.

NFTL - The number of unassigned tables in the FTL.

FTL - Free Table List. The list of all unassigned tables available for use. The tables are deleted from this list as needed by GTABZ. Tables no longer needed by a picture are returned to the list by CPICZ or PUTZZ. Note that space is allocated for NTABL tables, the maximum available in any given run is determined by the size of the work space assigned by the user in the call to INITZ.

PICID - Picture use table. Each word contains a displacement to a unique PPT. This table is used to assign a PPT to a picture name. When a PPT is assigned, the corresponding word in this table will contain a busy flag.

PPT - Picture Pointer Table Area. This area contains NPICT tables of 20 words each. A PPT is used to define a picture's status and characteristics. Status information is contained in the first four words and contains pointers to the data tables and defines the 'state' of a picture. The remaining 16 words contain all the information necessary to transform the user data into CRT code.

Figure 4 shows the format of the PPT status area. An inactive picture is defined as a picture temporarily stored on disk. This is designated by an $FF00_{16}$ in word zero of the PPT. The first disk sector address of the picture is saved in word one. The only action that may be taken on an inactive picture is to read it into core, clear it, or free it.

An active picture will have the first data table address stored in word zero of the PPT. Word one will contain the address of the last table in the chain, the 'active' table. This is the table currently being built. The relationship between the picture name, the PPT and data tables is demonstrated in Figure 5. The user's picture name points to the first word (word zero) of its PPT. The first word (word zero) of the PPT points to the first word in the first data table. The last word in each data table points to the next data table. The second word (word one) in the PPT points to the last data table. The last two words of the status area of the PPT contain internal status information not needed for a user understanding of interactive graphics operation.

The last sixteen words in the PPT are the characteristics section. These words are the characteristics of a picture, of (scaling, origin, intensity and density), all the information necessary to transform user data into CRT code. The following table contains a description of each of the words in this section of the PPT (see Figure 4):

<u>Words</u>	<u>Contents</u>
4-5	X axis expansion factor (see subroutine FACTZ). Initial value is 1.0.
6-7	X coordinate of picture origin in user units.
8-9	Number of raster units per user unit in X direction.
10-11	Y axis expansion factor. Initial value is 1.0.
12-13	Y coordinate of picture origin in user units.
14-15	Number of raster units per user unit in Y direction.
16	X accumulator. This is the X coordinate in raster units of the last point added to the picture. This coordinate is relative to the picture origin and has an initial value of zero.
17	Y accumulator. Y coordinate of last point in picture as above.
18	Current density and intensity values for the picture. Bits 13 - 15 contain the intensity, 8 - 12 the density, and 3 - 7 the density times 0.8. Initial values for both density and intensity are 5. Complete initial word value is 042D ₁₆ .
19	The pulse counter value for this picture. This is the current count of data words in the picture tables which actually contribute to the display.

ONLST - The 'on'list. This is a list of the PPT addresses of all pictures currently turned on. Only pictures turned on may be displayed. Note that the pictures turned on are a subset of the active pictures and the active pictures are a subset of all defined pictures.

CRT COMMUNICATIONS AREA (CCA)

NPICT EQU 6
NTABL EQU 16
PPTL EQU 20

LENGTH (in words)

1	USERZ - Address of user error routine.
1	NFTL - Number of free tables available.
NTABL	FTL - Free Table List.
NPICT	PICID - Picture use flags.
NPICT*PPTL	PPT - Picture Pointer Tables.
NPICT	ONLST - List of pictures turned on.

FIGURE 3

PICTURE POINTER TABLE (PPT)

WORD	CONTENT	INITIAL VALUE
0	STATUS AREA	
1		
2		
3		
4	X _f	1.0
5		
6	X ₀	0.0
7		
8	X _D	1.0
9		
10	Y _f	1.0
11		
12	Y ₀	0.0
13		
14	Y _D	1.0
15		
16	X ACCUMULATOR	0
17	Y ACCUMULATOR	0
18	$\frac{4}{5}$ DEN DEN INT	0
19	PULSE COUNTER	0

PPT STATUS AREA

0	Address of first table in chain.	
1	Address of last table in chain. (the active table)	
2	Not used	Picture ID
3	FF00 ₁₆ - if Picture is on. 0000 ₁₆ - if Picture is off.	

Picture active

FIGURE 4

0	FF00 ₁₆ - inactive	
1	DSA pointer to first disk sector containing this picture.	
2	Disk sector count.	Picture ID
3	Number of words used in last table.	

Picture inactive (on disk)

CRT PICTURE LINKAGES

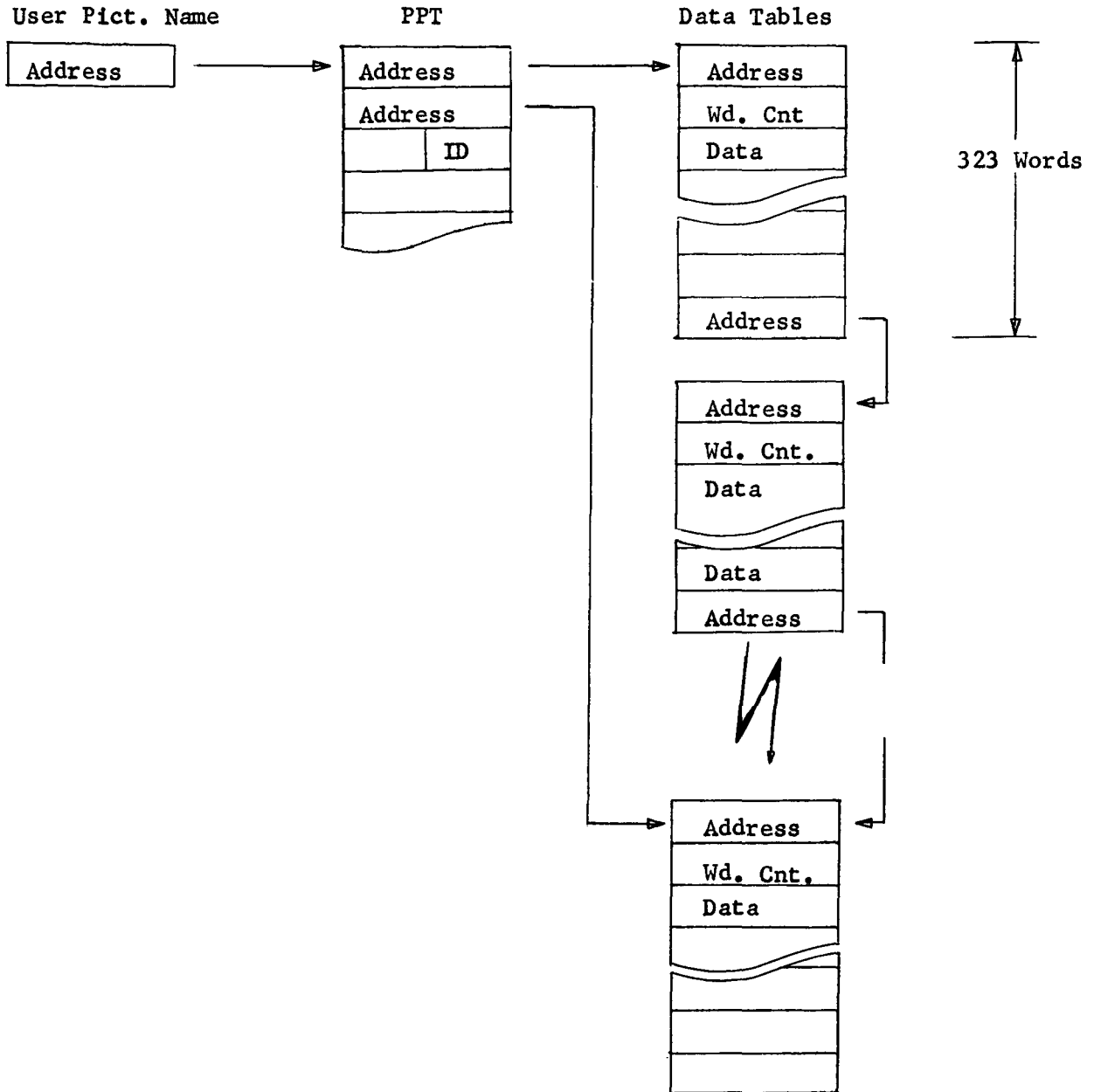


FIGURE 5

3.0 References

Bryden, Joseph E., "Design Consideration for Computer Driven CRT Displays", Computer Design, March 1969, pages 38-46.

IBM System/360: Operating System Graphic Programming Services for FORTRAN IV, Form GC27-6932.

Sherr, Solomon, Fundamentals of Display System Design, New York, John Wiley and Sons, 1970.

Woods, O. F., Cathode Ray Tube and Photo Pen Control Attachment with IBM Data Acquisition and Control System Manual of Instruction RPQ C08477, IBM Corp. San Jose, Cal., June 1971.

X-Y Display 1300A Operating and Service Manual, Hewlett-Packard Co., Colorado Springs Division, Colorado Springs, Colorado, June 1968.

4.0 Appendices

4.1 Appendix A - CRT ERROR MESSAGES

CRT error messages are generated by the routine ERRZ and are written in the form:

Zxx CRT yyyyyy ssssss AAAA QQQQ 1111 2222 3333 IIII UR

xx is the CRT error number in hexadecimal. yyyyyy is a term that indicates the general source of the error. If the picture name is available, it is stored in the accumulator (AAAA). The instruction register (IIII) contains the address of the call to ERRZ. On a user response error, the picture name, if it is available, is stored in word 145 of INSKEL COMMON.

Disposition in the following messages, refers to the action taken by ERRZ after writing a message to the user. The disposition codes are as follows:

W warning message only, processing will continue.

T terminal error, processing discontinued.

U user response error, ERRZ will call users error

response routine defined in the call to INITZ. The user must either correct the problem, or terminate the job. If the user does not provide an error recovery routine, he must use the system routine ERROR which defaults a user error to a terminal error.

xx	yyyyyy	DISP	DESCRIPTION
Z01	SPACE	T	Not enough work space. The array specified by the user in a call to INITZ is less than 323 words in length.
Z02	PICTRS	T	No free pictures available. The user has attempted to define more pictures than are available.
Z03	TABLES	U	No free tables available. A request was made for a free table and the FTL was empty. The user may respond by taking some action which will free one or more tables to the FTL.
Z04	UNDEF	W	Undefined picture. An attempt has been made to use an undefined picture. The call is ignored.

xx	yyyyyy	DISP	DESCRIPTION
Z05	FDA	U	No Free Disk Area available. An attempt was made to put a picture on disk and no FDA was available. The user may respond by 'reading' another picture from disk to core, thus making that DA available to this picture.
Z06	I/O	T	GETZZ/PUTZZ I/O error.
Z07	STATUS	W	Picture status. An attempt was made to perform an operation on a picture inconsistent with its status. The picture is inactive and the user attempted to manipulate it, or the picture is active and the user attempted to 'read' it from disk. The call is ignored.
Z08	DISK	T	No elements in CRTZZ. The 1810 Disk Working Storage file allocated to CRTZZ is shorter than the FTL. No FDAs can be defined, no pictures can be put on disk.
Z09	TABLES	U	Not enough data tables. An attempt was made to get a picture from disk and not enough data tables were in the FTL to contain the picture. The user may respond by performing any operation that returns data tables to the FTL.
Z0A		W	Indicates that a call was made to one of the intensity modification subroutines with an intensity of zero specified. No action is taken.
Z0D		W	A call to DLARY has been made with an illegal Fortran index (zero or negative). No deletion is made.
Z0E		W	An illegal table linkage was encountered in DLETZ or INT3Z. Chaining addresses did not agree.

4.2 APPENDIX B

SUBROUTINE REFERENCES

<u>SUBROUTINE NAME</u>	<u>SUBROUTINE REFERENCED</u>
CPICZ	BINHX, ERRZ, FMTZ, HOLPR, INITZ, PRT, PUTZZ
DLARY	BINHX, ERRZ, FMTZ, HOLPR, INITZ, PRT
DLETZ/INT3Z/INDXZ	BINHX, ERRZ, EXCRT, FMTZ, HOLPR, INITZ, PRT
DPICZ/FPICZ	BINHX, CPICZ, ERRZ, HOLPR, INITZ, PRT, FMTZ, PUTZZ
ERRZ	BINHX, EACPT, FMTZ, HOLPR, INITZ, PRT, ERROR (or user designated routine)
EXCRT	BINHX, ERRZ, FMTZ, HOLPR, INITZ, PRT
FLPTZ	BINHX, CPICZ, DPICZ, ERRZ, FADD, FALOG, FARC, FAXI, FDIV, FLD, FMPY, FMTZ, HOLPR, INITZ, IRNDF*, ISTOX, LDFAC, PRT, PUTZZ, SCALZ, SNR, SUBIN, SUBSC, XMDS
FMTZ	none
INITZ/ONZZ/OFFZ/GTABZ	BINHX, ERRZ, FMTZ, HOLPR, PRT
INTGZ	BINHX, CPICZ, DPICZ, ERRZ, FADD, FARC, FDIV, FLD, FMPY, FMTZ, HOLPR, INITZ, IRNDF*, ISTOX, LDFAC, PRT, PUTZZ, SCALZ, SUBIN, SUBSC
LIPNP	BINHX, CPICZ, DPICZ, ERRZ, FADD, FARC, FDIV, FLD, FMPY, FMTZ, HOLPR, INITZ, IRNDF*, PRT, PUTZZ, SCALZ
LITPN	none
PUTZZ/GETZZ/FDSAZ	BINHX, ERRZ, FMTZ, HOLPR, INITZ
RSETZ/DEN2Z/INT1Z/INT2Z/FACTZ/ GTXYZ	BINHX, ERRZ, FARC, FDIV, FLD, FMPY, FMTZ, HOLPR, INITZ, PRT
SCALZ/PLOTZ/MOVEZ/ALPHZ/RSCLZ	BINHX, CPICZ, DPICZ, ERRZ, FADD, FARC, FDIV, FLD, FMPY, FMTZ, HOLPR, INITZ, IRNDF*, PRT, PUTZZ
COMDP	BINHX, DMPHX, ERRZ, FMTZ, HOLPR, INITZ, PRT

* IRNDF is a locally written system subroutine which rounds the floating point number in the FAC and returns the integer equivalent in the accumulator.

4.3 APPENDIX C

SUBROUTINE STORAGE REQUIREMENTS

CPICZ	168
DLARY	68
DLETZ/INT3Z/INDXZ	112
DPICZ/FPICZ	144
ERRZ	184
EXCRT	130
FLPTZ	316
FMTZ	14
INITZ/ONZZ/OFFZ/GTABZ	342
INTGZ	174
LIPNP	220
LITPN	198
PUTZZ/GETZZ/FDSAZ	354
RSETZ/DEN2Z/INT1Z/INT2Z/ FACTZ/GTXYZ	196
SCALZ/PLOTZ/MOVEZ/ALPHZ/ RSCLZ	722
COMDP	15

TOTAL = 3357 Words

4.4 APPENDIX D

Indexed Data Arrays

If certain conventions are observed the photopen may be used to identify points in a data array by returning their indices to the user's program. The data must consist of a set of individual data points (i.e., the fourth parameter of the PLOTZ calls creating them must be one) which are determined by consecutive pairs of X and Y coordinates. These data points must be the first information displayed in the individual picture. If any information precedes the data array in this picture the returned index will be incorrect.

When such a data point has been trapped with the photopen and one of the three routines INT3Z (point intensity change), DLETZ (point deletion) or INDXZ (point identification) is called the FORTRAN index of that point will be returned in LIND (word 142 of INSKELE COMMON). This index may then be used in the user's program, or may be passed to subroutine DLARY which will remove the indexed element from each of an arbitrary number of floating point data arrays specified in the calling statement.

4.5 Appendix E

Basic Program Structure

The steps given below are necessary to the writing of a CRT display program. The sample program is shown in figure 6.

(a) The EXTERNAL ERROR statement must be the first statement in the program.

(b) An integer array must be allocated for use as a data table area. The dimension should be an integral multiple of 323. One data table will be available for each 323 words of the array.

(c) If the photopen interrupt feature is to be used, the INSKEL COMMON statement shown must be included, immediately following the last DIMENSION statement. IFILL is a dummy array used for positioning, and has no other significance in the program.

(d) A call to subroutine INITZ must precede all other CRT programming.

(e) If the photopen interrupt feature is to be used, a call to LIPNP must immediately follow the call to INITZ. The picture name assigned to the function matrix may not be used in any other CRT programming.

(f) A call to SCALZ must be made for each picture used in the display (except that of the function matrix.) This call must be made before any other programming is done for that picture.

(g) After all currently desired programming for a picture has been completed, a call to ONZZ must be made in order to include that picture in the display.

(h) A call to EXCRT must be made to initiate the display on the CRT tube.

```

// JOB      X      X      X      03 FEB 72 22.964 HRS
// FOR TEST  03 FEB 72 22.964 HRS
*LIST SOURCE PROGRAM
*INCS(1443 PRINTER)

(a) → EXTERNAL ERROR
      DIMENSION X(10),Y(10),IA(3230) ← (b)
      DIMENSION Z(10)
(c) → COMMON/INSKEL/IFILL(145),LR,LY,LX,LID,LFP
      DATA X/0.,50.,50.,100.,100.,300.,100.,50.,0.,0./
      DATA Y/0.,0.,100.,100.,200.,200.,100.,100.,0.,0./
      DATA Z/150.,170.,200.,150.,140.,175.,162.,210.,130.,230./
(d) → 10 CALL INIT7(IA,3230,ERROR)
(e) → 11 CALL LTPNP(IP0)
(f) → { CALL SCAL7(IP1,0,0,0,0,100,0,1,0,0,0,100,0,2,0)
        CALL SCAL7(IP2,500,0,0,0,100,0,1,0,0,0,100,0,0,5)
        CALL SCAL7(IP3,0,300,100,0,200,0,1,0,0,0,100,0,1,0)
        CALL SCAL7(IP4,500,300,0,0,100,0,1,0,100,0,200,0,1,0)
        CALL MOVE7(IP1,4,0,0,0,0)
        CALL MOVE7(IP2,4,0,0,0,0)
        CALL MOVE7(IP3,4,0,0,0,0)
        CALL MOVE7(IP4,4,0,0,0,0)
        CALL MOVE7(IP4,0,0,0,0,1)
        CALL MOVE7(IP1,0,0,0,0,1)
        CALL MOVE7(IP2,0,0,0,0,1)
        CALL MOVE7(IP3,0,0,0,0,1)
        CALL MOVE7(IP3,0,0,2,5,0)
        CALL MOVE7(IP1,0,0,2,5,0)
        CALL MOVE7(IP2,0,0,2,5,0)
        CALL MOVE7(IP4,0,0,2,5,0)
        CALL MOVE7(IP4,0,0,0,0,1)
        CALL MOVE7(IP1,0,0,0,0,1)
        CALL MOVE7(IP2,0,0,0,0,1)
        CALL MOVE7(IP3,0,0,0,0,1)
        DO 1 I=1,10
1 CALL PLOT7(IP1,X(I),Y(I),1)
        DO 2 I=1,10
2 CALL PLOT7(IP2,X(I),Y(I),0)
        DO 40 KK=1,10
          XX=KK*30
          CALL PLOT7(IP3,XX,Z(KK),0)
          CALL PLOT7(IP4,XX,Z(KK),0)
40 CONTINUE
100 FORMAT('1',8I7)
      WRITE(3,100)IP1,IP2
(g) → { CALL ONZZ(IP1)
        CALL ONZZ(IP2)
        CALL ONZZ(IP3)
        CALL ONZZ(IP4)
(h) → CALL EXCRT(0)
      CALL EXIT
      END

```

Figure 6 Sample CRT display program